An Association-Rules Learning Approach to Unsupervised Classification of Street Networks

Quentin Goss*, Mustafa İlhan Akbaş†, Alireza Chakeri‡, Luis G. Jaimes*
*Department of Computer Science, Florida Polytechnic University, Lakeland, FL 33805
{quentingoss0323, ljaimes}@floridapoly.edu
†Electrical,Computer,Software&Systems Engineering, Embry-Riddle Aeronautical University, Daytona, FL 32114
akbasm@erau.edu
‡Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620
chakeri@mail.usf.edu

***Abstract*—Street networks (SN) are intricate connections of roads and intersections; there is an enormous amount of available, open source street network (SN) data, such as the worldwide SN hosted by OpenStreetMaps, a popular source of SNs used in SUMO (Simulation of Urban Mobility) models, including the TAPASCologne project (TAPASCP). These SNs are intricate and high in feature dimension, notable are the edges containing data about road name, number of lanes, speed limit, etc. As a result, SN models that utilize this data are robust and highly-detailed. In this paper, we identify five significant edge features (SEF) from exploratory data analysis of the TAPASCP, present an unsupervised methodology of classifying bags of edges though an association rules learning approach to SEF aggregation, and measure the success of our methodology with SUMO simulations.**

***Index Terms*—street networks, association rules learning, Apriori, transportation simulation, SUMO**

I. Introduction

The transportation system (TS) is the means of which people and cargo move around from point A to point B and consists of a street network (SN), some civil infrastructure such as bridges and traffic lights, and both public and private transport. As technology is introduced into a TS, it is considered an intelligent transportation system (ITS). ITSs are used in many field of research, such as vehicular crowdsourcing (VCS) mechanisms which consider regions of interest (ROI) located within an ITS, and the VCS participants travel along the SN of the ITS to move from source, to ROI, to destination.

Within VCS mechanisms, the participant models perform a bountiful amount of cost calculations during the decision of which ROI to visit – these often involve iterating through the potential routes of the vehicle, each ROI, and each destination. When the amount of vehicles, ROI, or edges, nodes, and other features of the SN is large, these calculations become high in computation cost.

One approach to improving the computational performance of a VCS mechanism is to reduce the complexity of the SN. A SN is a combination of edges E and vertices V that makeup a graph G where $G \leftarrow \{E, V\}$. By utilizing a discretized street network (DSN) generated from the original or unmodified street network (USN) in place of the USN, similar vehicle trajectories are achieved with less computational cost. During the process of DSN generation, the existence and direction of spatially close edges in the USN are captured by the edges of the generated DSN. A shortcoming of existing DSN generation techniques is the failure to consider the additional edge features during this aggregation progress.

In this paper, we utilize the TAPASCologne project (TAPASCP), SUMO, TraCI, Apriori, R, Python, and the DSN generation technique [1], as well the broad concepts identified in the literature review to provide the following contributions:

1) Identification of five significant edge features (SEFs) from EDA of the TAPASCP.
2) An unsupervised methodology for classifying bags of edges through an association rules learning approach to SEF aggregation.
3) Applying the methodology to a DSN generation process to improve the quality of the generated DSN.
4) Validation of the methodology through SUMO simulations.

In Section II, we examine related works, then in Section III, we perform exploratory data analysis (EDA) on the TAPASCP, then we describe our methodology in Section IV, followed up by a discussion of SUMO experiments and results to evaluate our methodology in Section V, and a conclusion in Section VII.

II. Related Work

Originating from Euler's work on the Könisburg seven bridges [2] in 1736, graph theory and network science has overlapped. Today, street-to-graph conversions are commonplace in applications with ITSs. This is seen in the generation of a traffic-considering ITS for an inter-campus university shuttle service [3], to optimize movement of taxis to minimize empty ride time in urban taxi systems [4], and to account for the effect of nearby traffic during travel time prediction.

The complexity and adaptability of utilizing SNs as graphs is prevalent in approaches to solving the shortest path problem in ITSs, such as a robust shortest path model utilizing partial information of travel time distributions [5], in the comparison and parallel implementation of genetic shortest path and ant

978-1-7281-6861-6/20/$31.00 ©2020 IEEE

colony algorithms [6]. Graph and network properties may also exceed two dimensional constraints in an ITS to optimize other vehicular systems such aerial [7] or sensor networks [8].

The discretization of SN components of an ITS is commonplace, and is used at a varying degree of granularity. Navigation maps for intersection crossing are identified through discretized Gaussian process to identify merging and crossing area [9], and for the planning of merging scenarios [10]. In one case, roads (edges), are discretized into segments and lanes of equal size for traffic utilization analysis and prediction [11]. A low-detail example, is the discretization of the Anhui expressway into traffic accident segments [12]. The existing DSN generation methodologies are narrow and specific focusing on one component such as an intersection [9], a small SN with few edges [11], or an intersection and a short length of road approaching and leaving [10]. In another case, the purpose is for data collection rather than simulation [12]. There is a single DSN generation methodology for a SN as a whole, that produces a discretized street network (DSN) with grid-like properties and almost uniform distance between vertices. This approach accurately preserves spatial and length properties of a SN [1].

SN datasets and software tools for utilizing SN data are widely available. A notable SN database is Open Street Maps (OSM) [13] – OSM is a common source of SNs for software such as OSMnx [14] which retrieves and visualizes SNs, AOP [15] which increases the quantity of vertices of SNs to improve pedestrian movement accuracy, and SUMO (Simulation of Urban Mobility) [16] which is a traffic modeling and simulation (M&S) software. SUMO is open source, operates on OSM SNs, stores SNs in an easy to understand and modify format, has tools for generating custom SNs, options for in-depth data collection, and a Python 3.x interface TraCI (Traffic Control Interface) [17] that enables vehicular M&S. TAPASCP [18] is an advanced mobility dataset consisting of a SUMO SN that is retrieved from OSM that have been checked and modified for accuracy. Included are two prepared scenarios of twenty-four hours of traffic in Cologne, Germany, and a shorter version from 6:00am-8:00am.

Data mining makes an appearance in ITSs, such as unsupervised clustering of ITS data [19], analyzing traffic by generating association rules [20] using the Apriori algorithm [21] for association rules learning (ARL), and for decision analysis on ITS improvements based on big data mining [22].

There exists one DSN generation methodology which discretizes an entire SN. The shortcomings of this DSN generation methodology is the discretized edge aggregation, which only considers spatial edge data. Our contribution addresses these shortcomings with our edge feature aggregation (EFA) methodology. Additional to solving the shortcomings of edge aggregation in [1], our methodology is general an application to the categorization of any bag of SN edges. The ARL process of our EFA methodology differs from the existing data science work in ITSs as we utilize SEFs identified during EDA used in our EFA methodology, which are unique to this research.

III. Exploratory Data Analysis

To understand the properties of SN edges, we perform our EDA on the 71,368 edges of the TAPASCP, we select the TAPASCP as the SN dataset for this research because it is a robust and credible work with many citations and a statistically significant amount of edges.

A. Relevant Edge Features

The unique identifiers (IDs), start and end vertices IDs, and name, are irrelevant features for categorization; there is not enough data for road and sidewalk width to be significant. Of the set of all edge features, the five presented in Table I are found to be significant as they are present in all edges of the SN and are categorical. The three numerical features,

TABLE I: Significant Edge Features

Feature	Type*	Description
numLane	NC	Number of lanes on one side of the road.
priority	NC	Value used to resolve same-time-vehicle-arrival conflicts at when edges meet.
speed	NC	Speed limit in meters per second.
spreadType	C	The lanes vehicles default to when entering the edge {"right","center"}
type	C	Classification of Road (Highway, Residential, etc.)

*N = Numeric, C = Categorical

speed, *numLanes*, and *priority* fall into a small amount of buckets, shown in Figure 1 therefore, those feature also have categorical traits. The feature of *length* is used as weight for

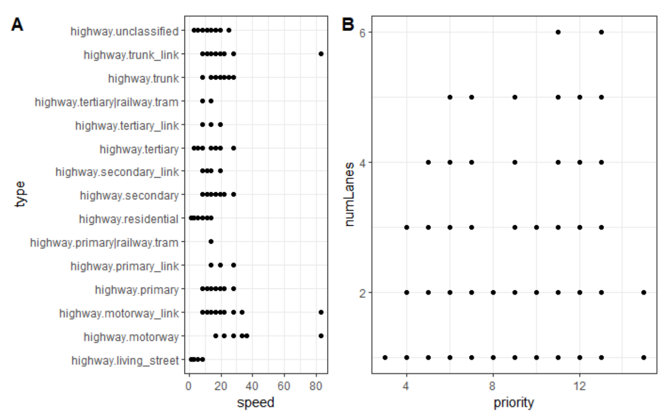


Fig. 1: The features *speed*, *priority*, and *numLanes* are both numerical and categorical.

the edges during ARL. When building the set of all edge data $e \forall e \in E$, the amount of contribution c_e for any e is $c_e = \text{floor}(\frac{\text{edge.length}}{8m}) + 1$ and edge.length is in meters. $8m$ is the optimal discretization increment of edge length for the TAPASCP as it is the discretization increment used in [23] on the same dataset.

B. Correlation and Principle Component Analysis

Principle component analysis (PCA) is coded and visualized with R [24] and the R package *caret* [25]. Three principle components (PCs) are generated (Table II). The biplot (Figure

2), and the correlation plot (Figure 3) of the numerical features reveals a strong correlation between *priority* and *numLanes*, and a moderate correlation of *speed* and *numLanes*, and *speed* and *priority*. The strength of the correlations revealed from

TABLE II: Importance of Components. PCA of the three numerical features.

	PC1	PC2	PC3
Standard Deviation	1.4982	0.7317	0.46903
Proportion of Variance	0.7482	0.1784	0.07333
Cumulative Proportion	0.7482	0.9267	1.00000

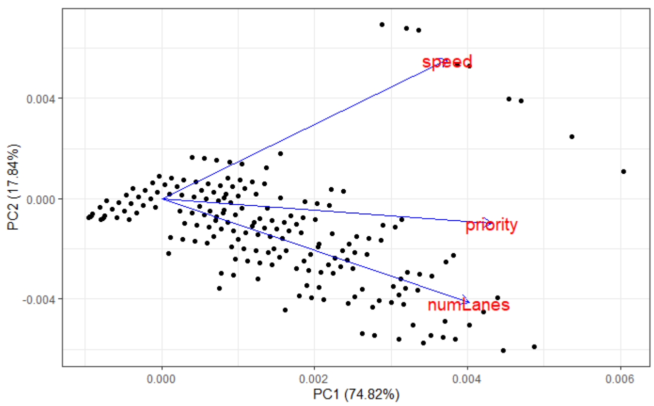


Fig. 2: Biplot of numerical features.

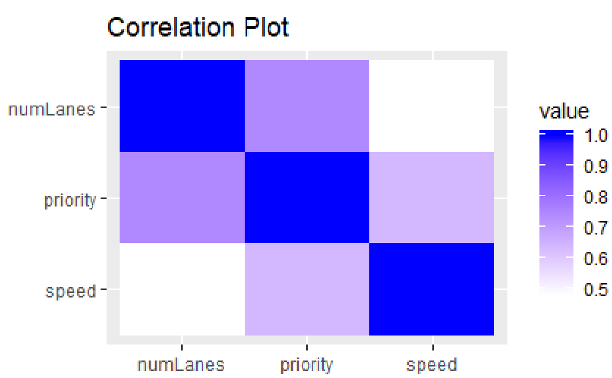


Fig. 3: Correlation between numerical features.

PCA and correlation analysis imply that these three features should be part of the generated edge aggregation process of DSN generation.

C. Association Rules Learning

ARL has been an efficient method of classification in data and computer science since the premier of the Apriori algorithm by Agrawal and Sirikant in 1994 [21]. Since then, efficient adaptations of Apriori and visualization tools thereof have been introduced, such as the *arules* [26] and *arulesViz* [27] R packages, that we use in our approach. We perform ARL on $e \forall e \in E$ of TAPASCP considering the SEFs of Table I with a minimum length of 5. This results in the generation of association rules (ARs) containing all SEFs. A summary of the first ten ARs is presented in Figure 4. The

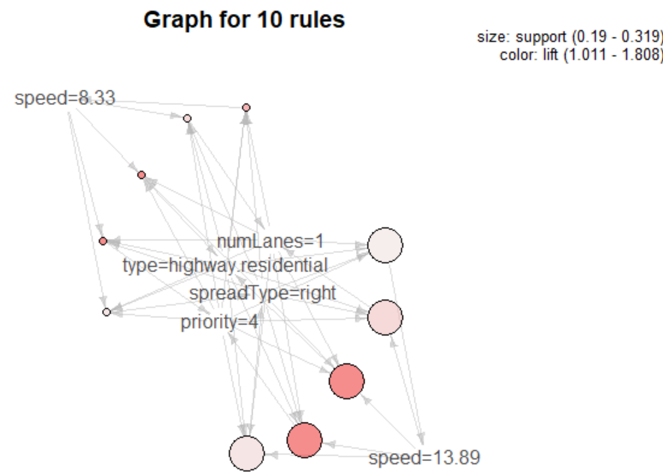


Fig. 4: Apriori association rules given support = 0.1, confidence = 0.3, and minimum length = 5.

information of Figure 4 reveals a classification of the SN with a concise summary and a granularity which includes every edge in the network. Figure 4 describes a suburb with two-lane roads (one in each direction), where the speed limit is mostly $13.89\frac{\text{meters}}{\text{second}}$ (i.e. $50\frac{\text{kilometers}}{\text{hour}}$ or $31\frac{\text{miles}}{\text{hour}}$) and occasionally $8.33\frac{\text{meters}}{\text{second}}$ (i.e. $30\frac{\text{kilometers}}{\text{hour}}$ or $18.6\frac{\text{miles}}{\text{hour}}$).

IV. Process

The edge feature aggregation (EFA) component is the primary contribution of our methodology in this section, which we explain first. To show the benefit of EFA, we also explain how to apply EFA to DSN generation to enhance the quality of generated DSNs.

A. Edge Feature Aggregation

Our technique will aggregate the features of any bag of edges E $\forall$ edges $e \in E$, thus the clustering methodology used to select E is irrelevant. The aggregated features are combined into a generated edge b, which is the classification of E. The EFA procedure is elaborated in Algorithm 1, which requires three input; a bag of edges E, as well confidence c, and support s. c and s are part of the ARL function call in Algorithm 1 line 26. In Section III, we identified five relevant, categorical edge features in the TAPASCP. These features are combined to form a transaction. The length of each edge determines the weight of the transaction in the ARL algorithm. This is achieved by increasing the number of transaction copies with the equation in Algorithm 1 line 18. The reason for the value of 8 is described in Section III-A.

1) Example 1: When E is a bag of all edges in the TAPASCP, $c = 0.1$, and $s = 0.3$, the most frequent rules, are displayed in Figure 4. Algorithm 1 selects the top rule, sorted by lift, then count. The resulting aggregation of features consist of *numLanes* = 1, *type* = highway.residential, *spreadType*

Algorithm 1 Generate a new edge with aggregated features using ARL.

```
input: Bag of edges e ∀ e ∈ E, Confidence c, Support s
output: Edge generated from aggregated edge features b
procedure EFA(E, c, s) begin
/* 1 edge in bag. C is a direct copy.                    */
if size(E) == 1 then
    return e0
end
/* Select categorical edge features.                     */
D ← Empty list to hold edges w/ only categorical features
n ← # of features
for all edges e ∈ E do
    d ← {
        type ← e.type
        spreadType ← e.spreadType
        speed ← string(e.speed)
        numLanes ← string(e.numLanes)
        priority ← string(e.priority)
    }
    /* Use length as weight, by increasing the #
    of occurrences in the edge bag.                      */
    x ← floor(double(e.length)/8) + 1
    while x < 0 do
        D.append(d)
        x ← x − 1
    end
end
T ← transactions(D) /* Create transactions.             */
/* Get association rules w/ any ARL algorithm            */
R ← Apriori(data=T,confidence=c,support=s,minLen=n)
/* Order the rules by lift then count.                   */
Order(R, descending, by={lift, count})
b ← { /* Select the first rule.                          */
    type ← R[0].type,
    spreadType ← R[0].spreadType,
    speed ← R[0].speed,
    numLanes ← R[0].numLanes,
    float ← R[0].priority
}
return b
end procedure
```

= right, *priority* = 4, and *speed* = 13.89. The output edge b consists of the aforementioned features.

2) Example 2: In [1], the generated edges of the DSN are a summary of USN edges around the (x, y) point of the DSN vertices. This summarization is graphically shown in Figure 5, in which Figure 5B preserves shape data of Figure 5A. There are as many bags of edges as there are vertices in the DSN; each of these edges bags, $c = 0.1$, and $s = 0.3$ are input in Algorithm 1 which output the more characteristic edges of Figure 5C.

B. Applying Edge Feature Analysis to Discretized Street Network Generation

Our EFA methodology may be applied to any DSN generation methodology – the pseudo-code is described in Algorithm 2. Three inputs are required, any SN that is to be discretized,

Algorithm 2 Applying EFA to a DSN generation process.

```
input: Any street network (USN), Confidence c, Support s
output: A discretized street network (DSN+)
procedure applyEFA(USN,c, s) begin
DSN ← Discretize(USN)
EB ← Correlate the edges from the DSN with the USN using
any clustering method. Place into bags.
B ← An empty list to hold generated edges
for all bags of edges E ∈ EB do
    b ← EFA(E, c, s)
    B.append(b)
end
DSN+ ← Apply the generated edges B to the DSN.
return DSN+
end procedure
```

referred to as an unmodified street network (USN), as well as a confidence c and support s value for the ARL algorithm. Our EFA methodology is to be performed on many bags of edges throughout the USN. A realistic approach to determining how many bags, and which edges of the USN should be placed into those bags, is to generate a DSN from the USN. Part of DSN generation involves clustering edges a process which considers spatial properties of SN edges when assigning clusters, resulting in each edge of a DSN being the aggregation of one or more edges in the USN. The result are edges bags that collectively cover all edges in the USN. New edges b are generated with our EFA methodology. The edges of the DSN are updated with the new edges $b \,\forall\, b \in B$, resulting in DSN+.

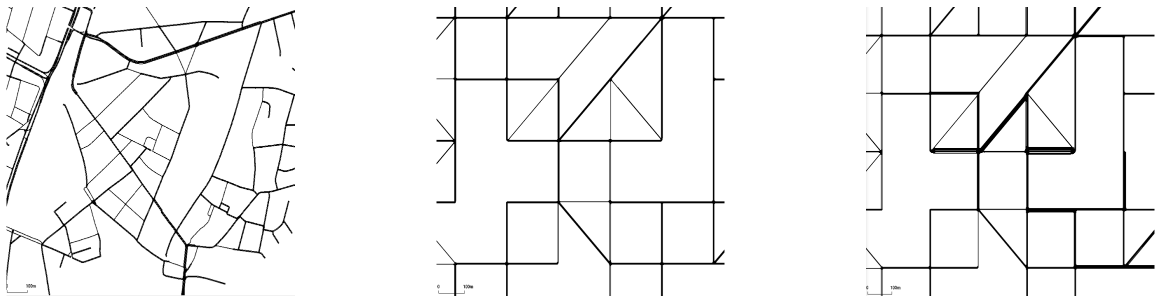
1) Example: A visualization of the input, middle, and output of Algorithm 2 is presented in Figure 5, depicting a zoomed in portion of the TAPASCP. USN = TAPASCP (Figure 5A), confidence $c = 0.3$, and support $s = 0.1$. A DSN is generated using the methodology of [1] (Figure 5B). The result is Figure 5C. Visually shown is a variation in the *numLanes* feature of the edge when comparing Figure 5C with B. Also applied to the DSN+ are the four non-visual SEF described in Table I.

V. Evaluation

We evaluate the performance of our methodology with three SUMO models:

- TAPASCP with 6:00am-8:00am trips (TAPASCP6-8)
- DSN of TAPASCP6-8 generated with [1] (SUMO-DSN)
- DSN of TAPASCP6-8 improved with our EFA methodology (SUMO-DSN+)

A trip in SUMO format includes an original edge ID, a destination edge ID, and a depart time. 109,254 trips from the USN of the TAPASCP6-8 are correlated with the SUMO-DSN and SUMO-DSN+. Trip information is generated by SUMO as output for each of the three simulation runs. We



(a) Unaltered TCP SN (USN). (b) Generated DSN using [1] (DSN) (c) DSN improved using ARL (DSN+).

Fig. 5: Comparison of generated DSNs (b,c) to the USN (a). Visual difference is in *numLanes*, non-visual is *priority*, *speed*, *spreadType*, and *type*.

evaluate the improvement of SUMO-DSN+ versus SUMO-DSN by measuring the difference of each trip information feature e.x. $\Delta\text{arrival}_{\text{SUMO-DSN+}} = $ TAPASCP6-8.arrival − SUMO-DSN+.arrival.

A detailed breakdown of measured features is presented in Table III which is the numerical summary of Figure 6. The % of Absolute Improvement is the measurement improvement of using SUMO-DSN+ versus SUMO-DSN, which is determined by Equation 1.

$$\% \text{ of Absolute Improvement} = \frac{|\text{SUMO-DSN+}|}{|\text{SUMO-DSN}|} * 100 \quad (1)$$

For example, to calculate the % of Absolute Improvement for the 3^{rd} Quarter of Table IIIA, SUMO-DSN = -53 and SUMO-DSN+ = 124. Thus $\frac{|-53|}{|124|} * 100 = \frac{53}{124} * 100 = 57.26\%$.

A. Measured Features

a) Arrival Time: Arrival time is the time in seconds when a vehicle reaches the end of it's trip. The boxes of Figure 6A are below 0, which means an early arrival time. Table IIIA confirms an improvement in arrival time on SUMO-DSN+. There is an improvement with early arriving, with a noticeable reduction in 1^{st} quarter of 22.25% and minimum by 33.57%.

b) Arrival Speed: Arrival speed is the speed in $\frac{\text{meters}}{\text{second}}$ that a vehicle is traveling when it reaches the end of it's trip, a negative value means a vehicle arrives at a speed slower than expected. The SUMO-DSN performs poorly (Figure 6B), whereas SUMO-DSN+ performs very close to TAPASCP6-8. We suspect that the poor behaviour of SUMO-DSN is the result of traffic jams since each edge is 1-lane – SUMO-DSN+ alleviates this congestion by considering *numLanes* during the aggregation process.

c) Duration: Duration is the amount of time in seconds that a vehicle takes to travel from the start to the end of it's trip, a negative value means a trip ends sooner than expected. This measurement of duration in Figure 6C and Table IIIC are similar to arrival time (Figure 6A and Table IIIA), in that the results show vehicles arriving early with the SUMO-DSN+ being an improvement over SUMO-DSN.

d) Route Length: Route length is the distance in meters that a vehicle travels from the start to the end of it's trip. The outliers in Figure 6D of route length samples are likely caused by vehicles being re-routed around congested regions at the time when the vehicles are added to the simulation, which is when route selection occurs in our experiments. From Table IIID, it is shown that the 3^{rd} quarter and Mean of the SUMO-DSN+ show improvement, while minimum and 1^{st} quarter results are less accurate. Since the mean improves by 40.39%, we consider the SUMO-DSN+ an overall improvement to route length.

e) Waiting Time: Waiting time is the amount of time in seconds that a vehicle is stopped, which occurs at intersections, traffic lights, or traffic congestion. Negative values means that vehicles wait less than expected. The results (Figure 6E and Table IIIE) reveal that the discretization process results in less than expected waiting time, likely caused by the reduction of intersections during the discretization process. The SUMO-DSN+ improves closeness at the minimum, 1^{st} quarter, and mean.

f) # of Times had to Wait: The count of times that a vehicle stops moving for any reason, negative values mean that less stops occurred than expected. Shown in Figure 6F and Table IIIF, are vehicles stopping less than expected on both the SUMO-DSN and SUMO-DSN+. The SUMO-DSN+ is closer to TAPASCP6-8 than SUMO-DSN with improvement at the minimum, 1^{st} quarter, and mean.

VI. Acknowledgments

This material is based upon work partially supported by the National Science Foundation (NSF) under NSF Award Number 1739409. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s), and do not necessarily reflect those of the NSF.

The street network data is partially derived from OSM data ©OSM contributors and available at www.openstreetmap.org/.

VII. Conclusion

In conclusion, we identify five SEFs from a credible SN dataset of 71,368 edges through EDA, then present an ARL

TABLE III: Summary of Trips Comparison

	A. Arrival Time (s)			B. Arrival Speed ($\frac{m}{s}$)			C. Duration (s)			D. Route Length (m)			E. Waiting Time (s)			F. # Times had to Wait		
	DSN+[1]	DSN[2]	%[3]	DSN+	DSN	%	DSN+	DSN	%	DSN+	DSN	%	DSN+	DSN	%	DSN+	DSN	%
Min	-33606	-50589	33.57%	-47.96	-55.54	13.65%	-33606	-49025	31.45%	-4917.2	-4279.1	-14.91%	-32519	-46458	30.00%	-177	-218	18.81%
1^{st} Quarter	-11039	-14194	22.23%	-10.07	-20.29	50.37%	-10328	-12513	17.46%	-382.7	-116.8	-227.65%	-8922	-11478	22.27%	-27	-33	18.18%
Median	-4147	-4577	9.39%	0.13	-15.48	99.16%	-3293	-3312	0.57%	99.3	324.5	69.40%	-2414	-2941	17.92%	-3	-4	25.00%
Mean	-6041	-8184	26.19%	0.57	-15.05	96.21%	-5489	-6927	20.76%	490.9	823.5	40.39%	-4603	-6430	28.41%	-9	-15	40.00%
3^{rd} Quarter	-53	124	57.26%	4.83	-12.53	61.45%	-32	146	78.08%	819.8	1141.9	28.21%	17	17	0.00%	4	4	0.00%
Max	24155	24455	1.23%	51.75	37.14	-39.34%	22333	23125	3.42%	28637.5	28637.5	0.00%	22153	22211	0.26%	215	226	4.87%

[1] SUMO-DSN+, [2] SUMO-DSN, [3] % of Absolute Improvement

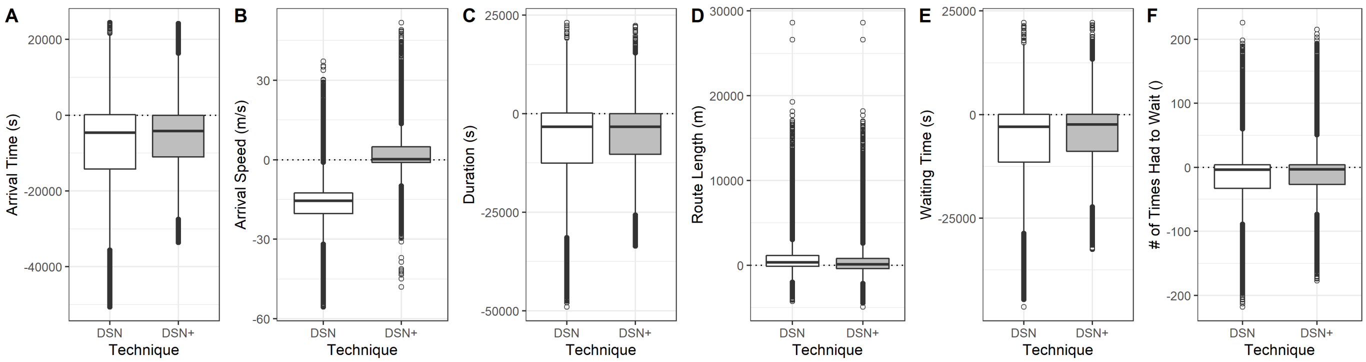


Fig. 6: Difference of Trip Information Features of discretization techniques VS TAPASCD6-8

approach for the unsupervised classification of SNs with an EFA methodology applicable to any bag of edges. In addition, we apply our EFA methodology to the domain of DSN generation, proving increases in quality to an existing DSN generation methodology.

The future plan for this work is to address the outliers in the difference of trip information (Figure 6), some of which are extreme. While this work improves the overall quality of DSN generation, going forwards it will be necessary to research methodologies to resolve or remove the outlier cases. Additional future work is to develop an aggregation methodology for the vertices of a SN, which would accurately handle same-time-vehicle-arrival conflicts and consider traffic signals along with their features. Other possibilities include research into alternative methodologies of bag of edge identification, or adaptations of our EFA methodology for non-edge SN components.

References

[1] Q. Goss, M. İlhan Akbaş, A. Chakerim, X. Wang, and L. G. Jaimes, "Gdtsn: Grid discretization technique for street networks," *IEEE International Conference on Communications (ICC)*, submitted for publication.

[2] L. Euler, "Solution problematis ad geometrian situs pertinentis," *Comm. Acad. Sci. Imp. Petropol*, vol. 8, p. 128, 1736.

[3] Q. Nelson, D. Steffensmeier, and S. Pawaskar, "A simple approach for sustainable transportation systems in smart cities: A graph theory model," in *IEEE Conference on Technologies for Sustainability (SusTech)*, 2018, pp. 1–5.

[4] X. Zhan, X. Qian, and S. V. Ukkusuri, "A graph-based approach to measuring the efficiency of an urban taxi service system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2479–2489, 2016.

[5] Y. Zhang, S. Song, Z.-J. M. Shen, and C. Wu, "Robust shortest path problem with distributional uncertainty," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1080–1090, 2017.

[6] G. Katona, B. Lénárt, and J. Juhász, "Compare ant-colony and genetic algorithm for shortest path problem and introduce their parallel implementations," in *IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 312–319.

[7] M. İ. Akbaş, G. Solmaz, and D. Turgut, "Actor positioning based on molecular geometry in aerial sensor networks," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 508–512.

[8] M. R. Brust, M. İ. Akbaş, and D. Turgut, "Multi-hop localization system for environmental monitoring in wireless sensor and actor networks," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 5, pp. 701–717, 2013.

[9] M. Barbier, C. Laugier, O. Simonin, and J. Ibañez-Guzmán, "Functional discretization of space using gaussian processes for road intersection crossing," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 156–162.

[10] N. Evestedt, E. Ward, J. Folkesson, and D. Axehill, "Interaction aware trajectory planning for merge scenarios in congested traffic situations," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 465–472.

[11] D. Tian, G. Wu, P. Hao, K. Boriboonsomsin, and M. J. Barth, "Connected vehicle-based lane selection assistance application," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[12] L. Du, G. Song, Y. Wang, J. Huang, M. Ruan, and Z. Yu, "Traffic events oriented dynamic traffic assignment model for expressway network: a network flow approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 1, pp. 107–120, 2018.

[13] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org , 2019.

[14] G. Boeing, "Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers Environment and Urban Systems*, vol. 65, pp. 126–139, 07 2017.

[15] P. Vitello, A. Capponi, C. Fiandrino, P. Giaccone, D. Kliazovich, and P. Bouvry, "High-precision design of pedestrian mobility for smart city simulators," in *IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[16] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[17] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: An interface for coupling road traffic and network simulators," in *Proceedings of the 11th Communications

and Networking Simulation Symposium, ser. CNS '08. New York, NY, USA: ACM, 2008, pp. 155–163. [Online]. Available: http://doi.acm.org/10.1145/1400713.1400740

[18] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, May 2014.

[19] S. Anand, P. Padmanabham, A. Govardhan, and R. H. Kulkarni, "An extensive review on data mining methods and clustering models for intelligent transportation system," *Journal of Intelligent Systems*, vol. 27, no. 2, pp. 263–273, 2018.

[20] L. Gao and S. Gao, "Traffic congestion analysis based on association rules," in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 1950–1954.

[21] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499. [Online]. Available: http://dl.acm.org/citation.cfm?id=645920.672836

[22] H. Liao, "Intelligent transportation decision analysis system based on big data mining," in *Journal of Physics: Conference Series*, vol. 1168, no. 3. IOP Publishing, 2019, p. 032002.

[23] X. Zhu, S. A. Samadh, and T. Yu, "Large scale active vehicular crowdsensing," in *IEEE Vehicular Technology Conference (VTC-Fall)*, Aug 2018, pp. 1–5.

[24] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2019. [Online]. Available: https://www.R-project.org/

[25] M. Kuhn, J. Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, the R Core Team, M. Benesty, R. Lescarbeau, A. Ziem, L. Scrucca, Y. Tang, C. Candan, and T. Hunt, *caret: Classification and Regression Training*, 2019, r package version 6.0-84. [Online]. Available: https://CRAN.R-project.org/package=caret

[26] M. Hahsler, S. Chelluboina, K. Hornik, and C. Buchta, "The arules r-package ecosystem: Analyzing interesting patterns from large transaction datasets," *Journal of Machine Learning Research*, vol. 12, pp. 1977–1981, 2011. [Online]. Available: http://jmlr.csail.mit.edu/papers/v12/hahsler11a.html

[27] M. Hahsler, "arulesViz: Interactive visualization of association rules with R," *R Journal*, vol. 9, no. 2, pp. 163–175, December 2017. [Online]. Available: https://journal.r-project.org/archive/2017/RJ-2017-047/RJ-2017-047.pdf